Polarity Classification of Blog TREC 2008 Data with a Geodesic Kernel

Stephan Raaijmakers & Wessel Kraaij
TNO Information and
Communication Technology
Delft, The Netherlands
{stephan.raaijmakers,wessel.kraaij}@tno.nl

Abstract

In this paper we describe the TNO approach to large-scale polarity classification of the Blog TREC 2008 dataset. Our participation consists of the submission of the 5 baseline runs provided by NIST, for which we applied a multinomial kernel machine operating on character n-gram representations.[1]

1 Introduction

The polarity task of Blog TREC 2008 consists of retrieving and ranking for each of a total of 150 topics (queries) the positive and negative opiniated documents in the test collection. TREC has made available 5 topic-relevance baseline runs, to which polarity classification or opinion finding techniques can be applied. This allows participants to focus on one aspect of the processing chain. In this contribution, we describe the result of applying the TNO polarity classification approach to these 5 baselines. We discuss the results of our submissions in section 5 and present conclusions and lessons learned in section 6. In the next three sections, we describe the data, our feature representation, and the general outline of our setup.

[1]This work was supported by the European IST Programme Project FP6-0033812. This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

2 Data and pre-processing

The TREC Blog06 collection, a 148 Gigabytes sample of the blogosphere, is the result of an eleven-week period crawl (December 2005-February 2006). Due to the automated crawling process, the dataset contains not only legitimate blog postings, but also spam, javascript, homepages and RSS feed material. The data itself consists of raw HTML, with a total of over 3.2 million documents. In order to train a classifier on these class-labeled web pages, these documents have to be cleaned up and converted to plain text, which is by far not a trivial task. Our HTML to text conversion strategy consists of a dedicated DOM-parser effectively stripping the larger part of HTML tags and javascript code. We combined this parser with the `html2text` Python script[2] in sequence: following our dedicated parser, we applied `html2text.py`. While this produced reasonably clean text, we found that in a lot of cases the output data still contained tags and programming constructs. We surmise that our results are to a large extent influenced by this imperfect data preprocessing.

3 Character n-gram representations

We opted for a character n-gram approach to the polarity classification task. For every training document, we generated word boundary transcending character n-grams from 2 up to 6 characters. That is, the transition between two consecutive words, including the white space char-

[2]Available from http://www.aaronsw.com/2002/html2text/

acter, is expressed as an n-gram. For the sentence 'This car really rocks' subword character bigrams and trigrams ('subgrams') are

$$\begin{array}{l}\texttt{th},\texttt{hi},\texttt{is},\texttt{ca},\texttt{ar},\texttt{re},\texttt{ea},\texttt{al},\\ \texttt{ll},\texttt{ly},\texttt{ro},\texttt{oc},\texttt{ck},\texttt{ks},\texttt{thi},\texttt{his},\\ \texttt{car},\texttt{rea},\texttt{eal},\texttt{all},\texttt{lly},\texttt{roc},\texttt{ock},\texttt{cks}.\end{array} \quad (1)$$

A bigram and trigram representation that spans word boundaries produces

$$\begin{array}{l}\texttt{th},\texttt{hi},\texttt{is},\texttt{s\#},\texttt{\#c},\texttt{ca},\texttt{ar},\texttt{r\#},\\ \texttt{\#r},\texttt{re},\texttt{ea},\texttt{al},\texttt{ll},\texttt{ly},\texttt{y\#},\texttt{\#r},\\ \texttt{ro},\texttt{oc},\texttt{ck},\texttt{ks},\texttt{thi},\texttt{his},\texttt{is\#},\texttt{s\#c},\\ \texttt{\#ca},\texttt{car},\texttt{ar\#},\texttt{r\#r},\\ \texttt{\#re},\texttt{rea},\texttt{eal},\texttt{all},\texttt{lly},\texttt{ly\#},\\ \texttt{y\#r},\texttt{\#ro},\texttt{roc},\texttt{ock},\texttt{cks}\end{array} \quad (2)$$

with `#` a whitespace indicator.

Every document is represented by a term vector consisting of L_1-normalized character n-gram frequencies. In our recent work (Raaijmakers and Kraaij, 2008); (Wilson and Raaijmakers, 2008); (Raaijmakers et al., 2008) we have found ample evidence for the informativity of character n-grams. In (Raaijmakers et al., 2008) we demonstrated for a large array of experiments that character n-grams are the most informative source of information compared to phonemes, prosody and word n-grams. These low-level features in fact implement a form of *attenuation* (Eisner, 1996): a slight abstraction of the underlying data that leads to the formation of string equivalence classes. For instance, words in a sentence will invariably share many character n-grams. Since every unique character n-gram in an utterance constitutes a separate feature, this produces string classes, which is a form of abstraction. Zhang and Lee (2006) investigate similar subword representations, called key substring group features. By compressing substrings in a corpus in a trie (a prefix tree), and labeling entire sets of distributionally equivalent substrings with one group label, an attenuation effect is obtained that proves very beneficial for a number of text classification tasks.

Aside from attenuation effects, character n-grams, especially those that contain word boundaries, have additional benefits. Treating word boundaries as characters captures microphrasal information: short strings that express the transition of one word to another. Stemming occurs naturally within the set of initial character n-grams of a word, where the suffix is left out. In addition, some part-of-speech information is captured. For example, the modals *could, would, should* can be represented by the 4-gram `ould`. Likewise, the set of adverbs ending in *-ly* can be concisely represented by the 3-gram `ly#`.

4 Geodesic kernels

Recent work on document classifiation has demonstrated the benefits of *geodesic kernels* (Lafferty and Lebanon, 2005): support vector machines that deploy geodesic distance measures on L_1-normalized data. L_1 normalization corresponds to normalizing the frequencies ($|\cdot|$) of a bag of events $D = w_1, \ldots, w_n$, where $\mid w_i \mid$ is the frequency of event w_i in D:

$$L_1(\{w_1, \ldots, w_n\}) = \{\frac{|w_1|}{\sum_i^n |w_i|}, \ldots, \frac{|w_n|}{\sum_i^n |w_i|}\} \quad (3)$$

L_1-normalization of data entails an embedding of this data into the *multinomial manifold* $\mathbb{P}^n$: an infinitely differentiable, curved information space that is isomorphic to the parameter space of the multinomial distribution. This information space has geodesic properties: it is locally Euclidean and globally curved. Distances between points therefore are best measured using locally Euclidean and globally geodesic distance measures. Technically, the multinomial manifold $\mathbb{P}^n$ is isometric to the positive portion of the n-sphere with radius 2, $\mathbb{S}_+^n$ (Kass, 1989; Lebanon, 2005):

$$\mathbb{S}_+^n = \{\phi \in \mathbb{R}^{n+1} : || \phi || = 2, \forall i, \phi_i \geq 0\} \quad (4)$$

by a diffeomorphism $F : \mathbb{P}^n \mapsto \mathbb{S}_+^n$:

$$F(x) = (2\sqrt{x}_i, \ldots, 2\sqrt{x}_{n+1}) \quad (5)$$

This allows for measuring distance with a kernel K between two vectors x, y in the space $\mathbb{S}_+^n$:

$$K(F(x), F(y)). \quad (6)$$

where the shortest path connecting these two points in hyperspace actually is a segment of a great circle.

Raaijmakers (2007) demonstrates that multinomial kernels based on geodesic distance are able to produce state of the art results for sentiment polarity classification tasks.

In the experiments reported in this work, we use a simple, hyperparameter-free multinomial kernel, the negative geodesic kernel K_{NGD} (Zhang et al., 2005):

$$K_{NGD}(x, y) = -2 \arccos \left(\sum_{i=1}^{n} \sqrt{x_i y_i} \right) \quad (7)$$

Notice that this kernel combines a *local*, Euclidean notion of similarity with a geodesic notion of similarity: the vector product expresses cosine similarity, and the inverse cosine the measurement of distance along a curve.

Expanding the TREC data to character n-grams leads to a huge expansion of data. Due to memory constraints of our systems, we took a random portion of training data of only 16% (amounting already to over 250 megabytes of training data).

4.1 Thresholding decision values

Support vector machines output decision values that either are discretized to binary classes (a negative value produces a negative class label, and a positive value a positive class label), or probabilities (e.g. (Platt, 1999)). We used the raw decision values for ranking the various positive and negative cases. We devised a simple threshold estimator that, on the basis of class distribution priors in the training data, determines the optimal threshold above which decision values should produce positive classes. Algorithm 1 performs a one-parameter sweep, fixing a decision value threshold that optimally approximates the *a priori* class distributions in the training data. We used this threshold to assign classified documents to the positive and negative classes, prior to ranking their respective decision values.

Algorithm 1 Threshold estimation for decision value discretization

Require: $\delta_{min}, \delta_{max}, \sigma$; D_{train} (decision values training data); P_-, P_+ (priors); θ (threshold); σ (step size)
$N_+ \Leftarrow 0; N_- \Leftarrow 0$
$\lambda \Leftarrow \delta_{min}$
while $\lambda < \delta_{max}$ **do**
  for each d in D_{train} **do**
    if $d < \lambda$ **then**
      $N_- \Leftarrow N_- + 1$
    else
      $N_+ \Leftarrow N_+ + 1$
    end if
  end for
  $N_+ \Leftarrow \frac{N_+}{|D_{train}|}$
  $N_- \Leftarrow \frac{N_-}{|D_{train}|}$
  if $\mid N_+ - P_+ \mid \leq \theta$ **and** $\mid N_- - P_- \mid \leq \theta$ **then**
    return λ
  end if
  $\lambda \Leftarrow \lambda + \sigma$
end while
Return λ

5 Results

In figures 1 and 2, the results for positive and negative queries are displayed, by plotting the difference of the produced MAP and R-PREC

values[3] and the reference values. As can be seen, the runs for the positive queries produce well above median scores for both MAP and R-PREC. Averaged over the 5 baseline runs, for the positive queries, a portion of 62.3% is equal to or above the reference median average precision. For the R-PREC scores for positive queries this portion is on average 70.5%. The R-PREC scores produced by the 5 positive baseline runs were all significantly[4] better than the median R-PREC reference scores. The average difference between produced R-PREC and reference R-PREC was +12.1%. For the negative queries, on average, 24.7% of all MAP scores produced were equal to or above the reference MAP values. For R-PREC, a much higher proportion of on average 57.8% scores was equal to or above median reference R-PREC. The averaged difference over all 5 runs for R-PREC compared with reference R-PREC was -1.7%. In 4 out of 5 runs, this difference was significant, its average amounting to a rather small -1.7%.

The percentages of deviations are listed in table 1, as well as the results of the Wilcoxon signed rank applied to the R-PREC results.

Task	% MAP	% R-PREC	MAP	R-PREC	W
POSITIVE QUERIES					
base1	+66.9	+75.7	26.2	21.3	+14.7
base2	+53.4	+60.8	19.4	15.4	+7.9
base3	+64.2	+73	24.1	19.4	+12.5
base4	+64.2	+71.6	23.8	19	+12.3
base5	+62.8	+71.6	24.8	19.7	+13.2
Average	+62.3	+70.5	16.2	11.5	+12.1
NEGATIVE QUERIES					
base1	+22.7	+61	8.7	4.5	-1.3
base2	+14.9	+49.7	6.7	3.5	-3.3
base3	+27.7	+58.9	7.7	4	-2.3
base4	+31.2	+59.6	8.3	4.5	-1.6
base5	+27	+59.6	8.4	4	=
Average	+24.7	+57.8	10	6.7	-1.7

Table 1: Percentage of queries with MAP scores above/below (+/-) median average precision; percentage of queries with R-PREC scores above/below median R-PREC; average MAP and average R-PREC scores for the 5 polarity baselines; Wilcoxon significance of the difference of the obtained score with the reference score ($p < .5$), as well as the difference of the average obtained score with the average reference score.

[3]Mean Average Precision and Precision at R (with R the number of relevant documents).

[4]All significance results were computed with the non-parametric Wilcoxon signed rank test, with $p < .5$.

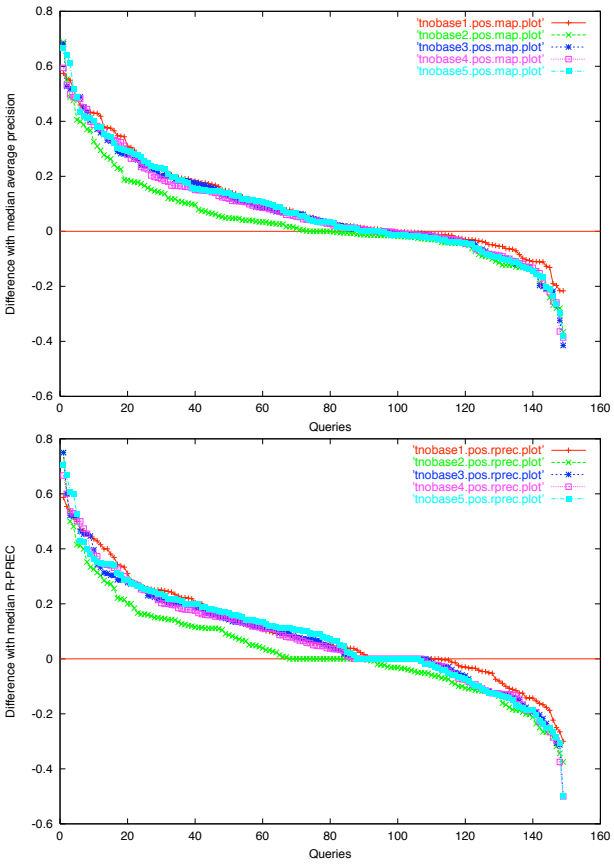

Figure 1: Difference of TNO produced MAP and R-PREC values with the TREC reference values for positive queries (queries sorted by descending performance).

6 Conclusions

In this paper, we presented the TNO approach to polarity classification and ranking of the Blog TREC 2008 data. For 5 baseline runs, we applied a geodesic kernel to character n-gram representations. We trained our system on a relative small portion of 16% of the total available training data. Results show that our system performs well above median for positive queries. For negative queries, results are in 4 out of 5 runs below median, albeit with a small (but significant) percentage. As a lesson learned, in future TREC participation, we will invest more time in thorough data cleaning prior to classifier training and testing.

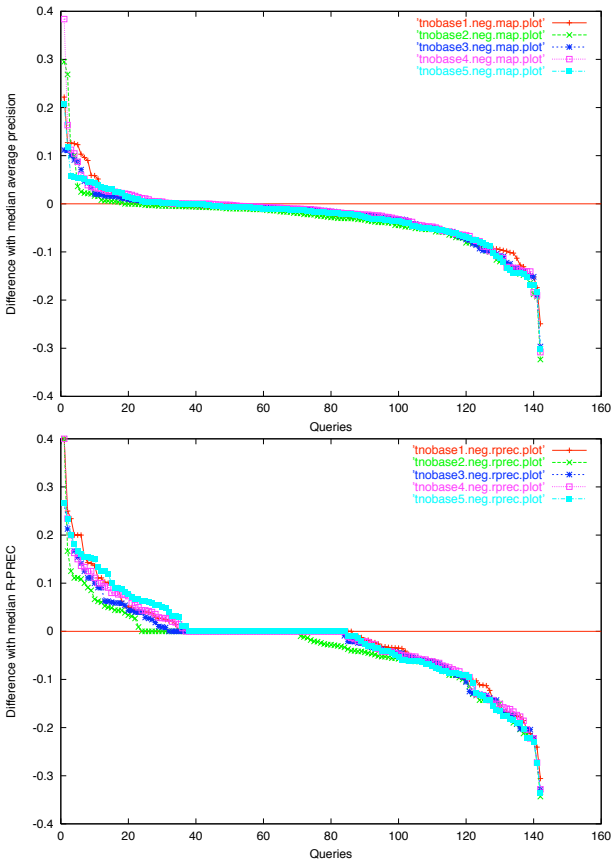

Figure 2: Difference of TNO produced MAP and R-PREC values with the TREC reference values for negative queries (queries sorted by descending performance).

References

J. Eisner. 1996. An empirical comparison of probability models for dependency grammar. In *Technical Report IRCS-96-11, Institute for Research in Cognitive Science, University of Pennsylvania*.

Robert E. Kass. 1989. The geometry of asymptotic inference. *Statistical Science*, 4(3):188–234.

John Lafferty and Guy Lebanon. 2005. Diffusion kernels on statistical manifolds. *Journal of Machine Learning*, 6:129–163.

Guy Lebanon. 2005. Information geometry, the embedding principle, and document classification. In *Proceedings of the 2nd International Symposium on Information Geometry and its Applications*.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74.

S. Raaijmakers and W. Kraaij. 2008. A shallow approach to subjectivity classification. In *Proceedings of ICWSM*.

Stephan Raaijmakers, Khiet Truong, and Theresa Wilson. 2008. Multimodal subjectivity analysis of multiparty conversation. In *Proceedings of EMNLP*.

S. Raaijmakers. 2007. Sentiment classification with interpolated information diffusion kernels. In *Proceedings of the First International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD'07)*.

Theresa Wilson and Stephan Raaijmakers. 2008. Comparing word, character, and phoneme n-grams for subjective utterance recognition. In *Proceedings of INTERSPEECH*.

D. Zhang and W. S. Lee. 2006. Extracting key-substring-group features for text classification. In *Proceedings of KDD*.

Dell Zhang, Xi Chen, and Wee Sun Lee. 2005. Text classification with kernels on the multinomial manifold. In *Proceedings SIGIR'05*, pages 266–273.